

Módulo III: SISTEMA SOPORTE DE SCHEDULING DE ÓRDENES DE PRODUCCIÓN

Pablo ROSSI¹, Clara DÍAZ¹, Pablo FRUTTERO¹, Mariela RICO¹, María R.GALLI² y Omar CHIOTTI³

(1) **GIDSATD - UTN** - Facultad Regional Santa Fe
Lavaisse 610 - 3000 SANTA FE - ARGENTINA

(2) **INTEC - CONICET** e-mail: mrgalli@intec.edu.ar

(3) **INGAR - CONICET** e-mail: chiotti@arcrade.edu.ar

RESUMEN

En este trabajo presentamos una descripción de un subsistema soporte de la actividad de scheduling de órdenes de producción en plantas de producción por lotes. Este subsistema forma parte de un Sistema Soporte de Decisiones global prototipo en desarrollo en el GIDSATD (Grupo de Investigación y desarrollo de Sistemas de Ayuda a la Toma de Decisiones). En particular, en primer lugar describimos brevemente la estructura diseñada para el manejo de la información necesaria para el scheduling, en segundo lugar presentamos un análisis y diseño orientado a objetos del sistema soporte de scheduling (SSS) y finalmente describimos el algoritmo de scheduling incorporado a la base de modelos.

INTRODUCCIÓN

Los permanentes cambios tecnológicos y la nueva visión de las empresas tendientes a satisfacer las expectativas del cliente, resultan en una considerable reducción del ciclo de vida de los productos y de los procesos productivos. Hoy las empresas ofrecen una gama muy amplia de productos; esto, sumado a la rápida obsolescencia de los mismos, está impulsando a las empresas a abandonar la tradicional política de inventarios de productos para pasar a una política de producción bajo órdenes de pedido.

Para ejecutar en forma competitiva una política de producción por órdenes se requiere un sistema que proporcione la información necesaria al momento de tomar decisiones. Estos sistemas deben ser provistos de una adecuada arquitectura para organizar la información y además deben incorporar modelos, estrategias y algoritmos para soportar la actividad de scheduling de las órdenes de producción. La necesidad de desarrollar este tipo de soporte informático ha sido planteada por muchos autores, quienes describen arquitecturas de sistemas diseñados para dar soporte a la actividad de scheduling. Algunos autores intentaron generar SSS a través de la integración de softwares que brindan distintos tipos de soporte. El resultado de esto, expresado por los mismos autores, es baja performance del sistema debido entre otros problemas a duplicación de cálculos e información y a incompatibilidades en la comunicación. A modo de ejemplo podemos citar el trabajo de Crooks y otros (1992). Por otra parte, encontramos publicaciones en las cuales se describen SSS diseñados a la medida de una empresa particular (por ejemplo, Adamopoulos y otros, 1994,). El problema de estos SSS es que generalmente son poco flexibles para ser adaptados a nuevas estructuras organizativas de la empresa o a otras empresas.

De este rápido análisis resulta evidente la necesidad de *integración*. A nuestro juicio, más que *integración*, creemos que existe la necesidad de soporte global de información. Esto es, cada área de una organización requiere de herramientas e información para tomar decisiones. Los modelos y herramientas generalmente son particulares de un área mientras que la información es global. Por lo tanto, un sistema soporte de decisiones competitivo requiere de una adecuada arquitectura y no puede surgir de la integración de softwares diseñados en forma independiente. Nuestro grupo está trabajando en el diseño de la arquitectura de un SSD global (Rico y otros, 1997). Con el término global nos referimos a un sistema que involucra todos los puntos de decisión de una organización y además que no es diseñado para una organización

en particular. En el desarrollo de este sistema, una de las actividades a soportar es la de scheduling de órdenes de producción.

En este trabajo describimos la arquitectura del subsistema soporte de scheduling (SSS) de órdenes de producción en plantas de procesamiento por lotes. En primer lugar describimos brevemente la estructura diseñada para el manejo de la información requerida para el scheduling, en segundo lugar presentamos un análisis y diseño orientado a objeto del SSS, y finalmente describimos la conformación de la base de modelos del SSS, la cual está compuesta esencialmente por los algoritmos de scheduling,

ESTRUCTURA DE INFORMACIÓN DEL SSS

La estrategia utilizada para estructurar la información en este sistema ha sido desarrollada en base al siguiente concepto: El objetivo final de una empresa de producción es el **producto**. El **producto** requiere **materiales** (materias primas, material empaque, etc.) y **actividades** (las actividades son todas las acciones que el producto insume para llegar al cliente en el tiempo y con la calidad deseada (Lorino,1993). Una **actividad** requiere de **recursos** (equipos y tiempo), **materiales** (no incorporados al producto), **servicios** (Energía eléctrica, vapor, agua de enfriamiento etc.) y **mano de obra**.

En base a esta conceptualización, desde el punto de vista de la información requerida para el scheduling, consideramos a la planta de producción como una *entidad capaz de realizar actividades*. Por ejemplo: A1, A2, A3, A4. A partir de este concepto de actividad, un *proceso de producción* se define como un conjunto de actividades con una relación de precedencia especificada. Por ejemplo, el proceso de producción $P1=\{A1,A2,A3,A4\}$ cuyas relaciones de precedencia vienen definidas por $RP(A1) = \phi$, $RP(A2) = \phi$, $RP(A3) = \{A1,A2\}$, $RP(A4) = \{A3\}$.

Para atender la solicitud de una orden de producción, la *actividad* podrá disponer de distintas alternativas de uso de los recursos para realizar la prestación. El algoritmo de scheduling no interviene en la organización interna de una actividad. Esto es, con la información del producto y la cantidad requerida por la orden un subsistema de *Capacidad de Producción* ayuda a definir el cronograma de operación de cada actividad para atender al proceso de producción correspondiente. Consideremos el ejemplo anterior, supongamos que se requiere producir n lotes para satisfacer la demanda de la orden, que además se elige la política de operar con el mismo tamaño de lote en todas las actividades, y que no hay procesamiento en paralelo en la actividad. El cronograma de operación de cada actividad vendrá dado por el siguiente modelo:

$$\begin{aligned} \{ \theta_{i,j,1} &= \max \{ Fd_i, \theta_{pi,j,1} + T_{pi,j,1} \} \quad \forall pi \in RP(i) \} \\ \theta_{i,j,n} &= \max \{ \theta_{i,j,n-1} + T_{i,j,n-1}, \theta_{pi,j,n} + T_{pi,j,n} \} \quad \forall pi \in RP(i) \quad \forall n > 1 \quad \forall i \in Pj \end{aligned}$$

Donde, $\theta_{i,j,1}$ es la fecha de inicio del procesamiento del lote 1 del producto j en la actividad i .

Fd_i es la fecha en que la actividad i está disponible para ser afectada a procesar la orden actual.

$T_{pi,j,1}$ tiempo requerido para procesar 1 lote de producto j en la actividad i.

Luego, para cada relación actividad-orden de producción, se tiene un cronograma de operación relativo a Fd_i . Este cronograma depende de una serie de factores:

- El producto a producir.
- La cantidad a producir.
- La política de lotes de producción (igual o distinto tamaño de lotes).
- La política de inventario de lotes en proceso (sin inventario intermedio, con inventario intermedio) entre dos actividades.

Sea $OR = \{\text{conjunto de órdenes a someter a scheduling}\}$, luego la tarea de scheduling implica definir el valor de $Fd_i \quad \forall i \in P_j$ y $\forall \text{Orden} \in OR$ en los modelos de la relación actividad-orden de producción, conformando así un cronograma de ejecución de las mismas.

ANÁLISIS Y DISEÑO ORIENTADO A OBJETOS

En esta sección describimos la estructura de clases del subsistema, la cual fue desarrollada a través de un análisis y diseño realizado siguiendo la metodología de objetos de G. Booch (1993).

Hemos realizado el análisis y diseño del módulo comenzando con una definición de los escenarios principales a los cuales el sistema debiera dar soporte. Como ejemplo presentamos el escenario principal, el cual denominamos *Scheduling*.

ESCENARIO : <i>Scheduling</i>
<ol style="list-style-type: none">1. El usuario requiere realizar un scheduling.2. Selecciona el conjunto de <i>órdenes</i> a someter a scheduling.3. Especifica el <i>objetivo</i> a satisfacer por el <i>schedule</i>.4. Especifica los <i>recursos</i> (<i>actividades, materiales, servicios y/o mano de obra</i>) a considerar en la selección del <i>schedule</i>.5. Selecciona el <i>algoritmo</i> de scheduling a utilizar.6. Define el <i>horizonte de tiempo</i> del <i>schedule</i>.

De este escenario, un análisis de sustantivos, permite inferir como clases/objetos candidatos los siguientes: *órdenes, schedule, objetivos, algoritmos, horizonte de tiempo, actividades, materiales, servicios, mano de obra*. En la figura 4 se muestra el Diagrama de Interacción correspondiente, el cual representa la interacción entre las clases/objetos candidatos.

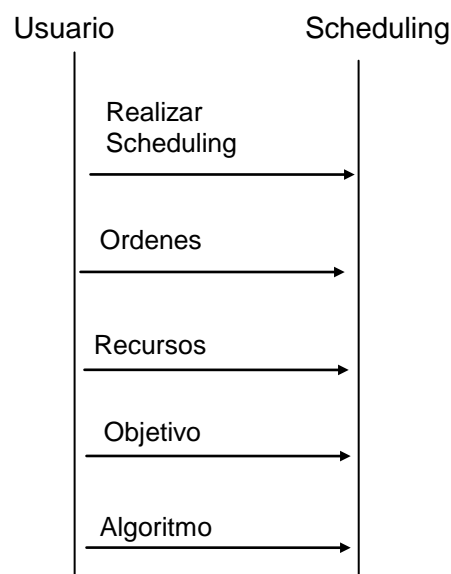


Figura 4: Diagrama de Interacción

En este trabajo utilizaremos el término “schedule” para referirnos al resultado de la tarea “scheduling”, la cual implica decidir un cronograma de ejecución de las actividades, un cronograma de requerimiento de materiales, un cronograma de requerimiento de servicios y un

cronograma de requerimiento de mano de obra, necesarios para generar los productos asociados a un conjunto de órdenes de producción.

Evaluando la definición del término *schedule*, es posible considerar a la clase/objeto *Schedule* como compuesta por las clases/objetos *Cronogramas de Actividades*, *Cronogramas de Materiales*, *Cronogramas de Mano de Obra* y *Cronogramas de Servicios*. El vínculo corresponde a una relación de composición.

También es posible inferir de esta definición una relación de composición de la clase *Schedule* respecto a la clase *Horizonte de Tiempo*. Estas relaciones se representan en el Diagrama de Clases 1.

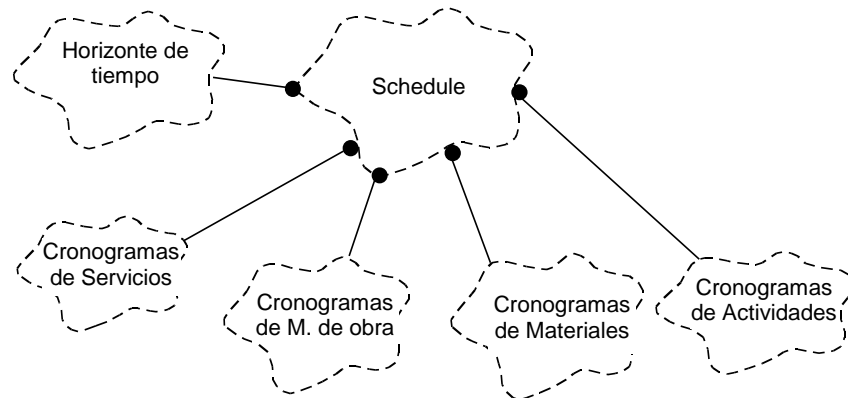


Diagrama de Clases 1

Por otra parte, existe una relación de asociación de la clase *Schedule* con la clase *Órdenes* y con la clase *Objetivos*. Esto es, un *schedule* necesariamente está asociado a un *objetivo*, en función de cuyo nivel de satisfacción fue elegido, y a un conjunto de n *órdenes* de producción especificado. Es decir, si el usuario cambiase el objetivo y/o el conjunto de órdenes, cambia el *schedule*. También existe una relación de asociación entre *Órdenes* y *Producto*, en particular cada orden está asociada a un único producto. Esto da origen al Diagrama de Clases 2:

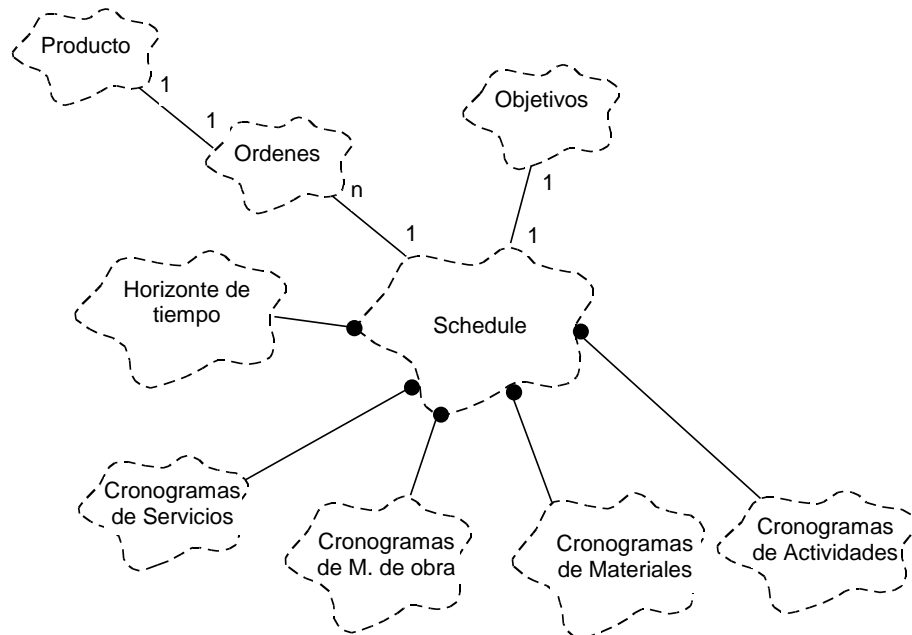


Diagrama de Clases 2

Para generar el *schedule* se utiliza el *algoritmo* de scheduling especificado por el usuario, por lo que la relación entre las clases *schedule* y *algoritmos* es de uso. Por otra parte, un algoritmo de scheduling en su proceso iterativo requiere determinar para cada solución en consideración el valor del objetivo especificado por el usuario, para esto hace uso de la método (función) asociada a dicho objetivo. Por lo tanto existe también una relación de uso de la clase *Algoritmos* respecto a la clase *Objetivos*. Estas relaciones se reflejan en el Diagrama de Clases 3.

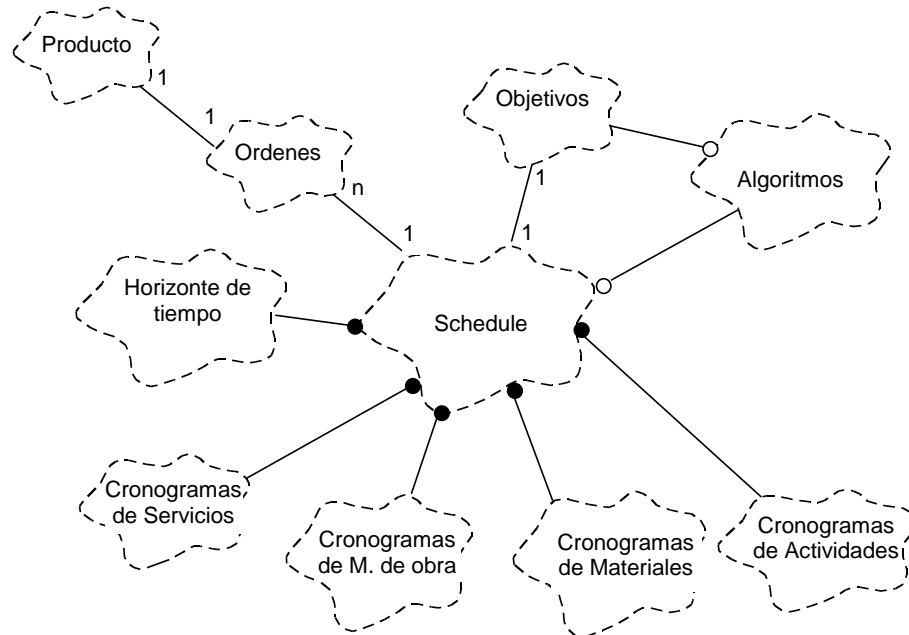


Diagrama de Clases 3

El cronograma de una actividad está asociado a dicha actividad. Se establece entonces una relación de asociación entre la clase *Cronogramas de Actividades* y la clase *Actividades* inferida inicialmente a partir del escenario *Scheduling*. Esta relación de asociación se establece también entre cada una de las restantes clases: *Materiales*, *Mano de Obra* y *Servicios*, inferidas también del escenario *Scheduling*, y las clases que contienen sus respectivos cronogramas. Representamos estas relaciones en el Diagrama de Clases 4.

En la sección anterior hemos definido el concepto de Actividad utilizado en este trabajo.

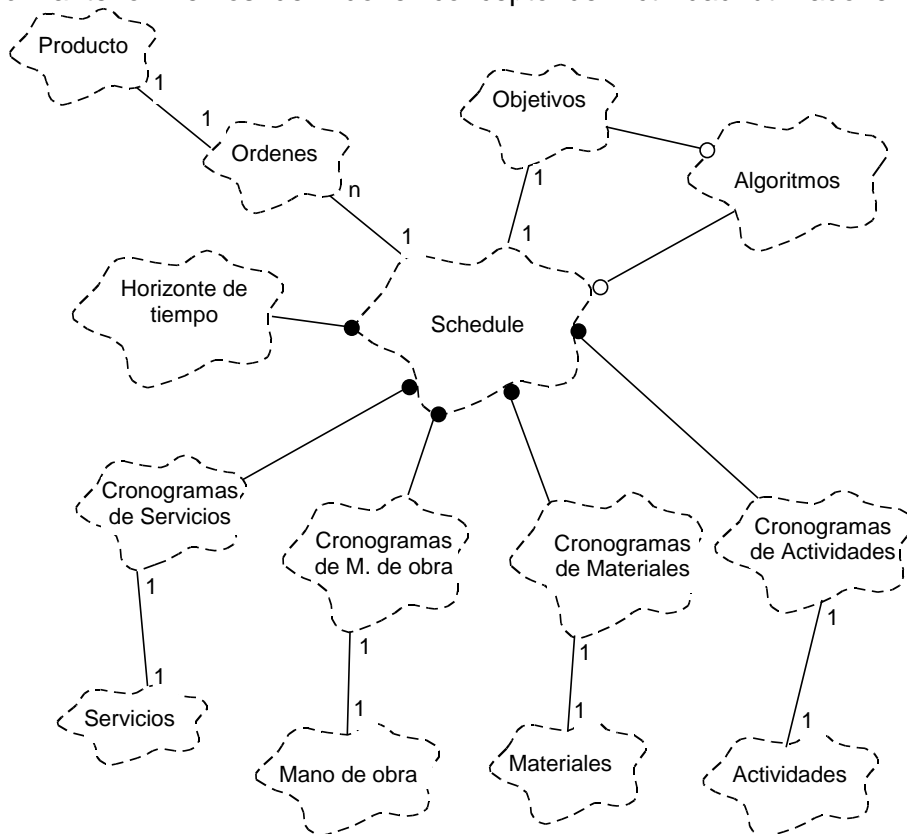


Diagrama de Clases 4

Además, hemos definido para cada actividad un modelo que determina el cronograma de operación interna de la misma. Luego, inferimos la existencia de una nueva Clase que denominamos *Modelos de Actividades*, la cual esta asociada a la clase *Actividades*.

El modelo que determina el cronograma de operación de una actividad es utilizado por el algoritmo de scheduling. Por lo tanto se establece una relación de uso de la clase *Algoritmos* respecto a la clase *Modelos de Actividades*. Representamos esto en el Diagrama de Clases 5.

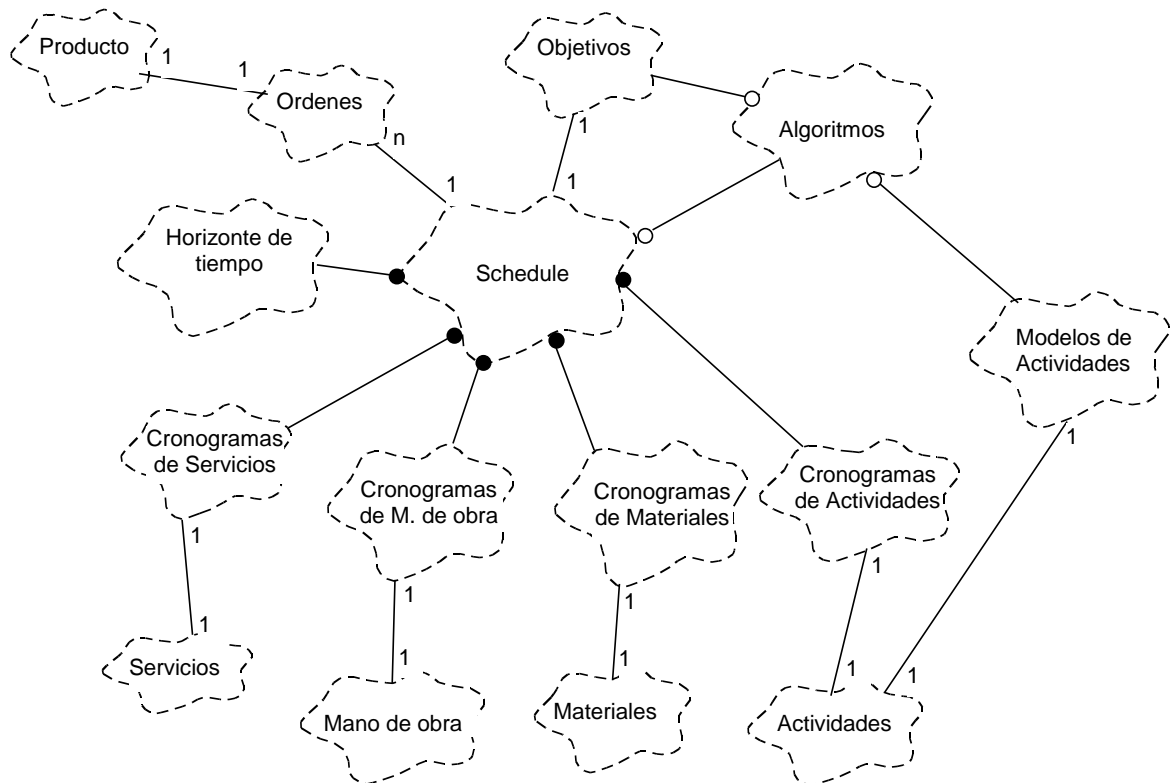


Diagrama de Clases 5

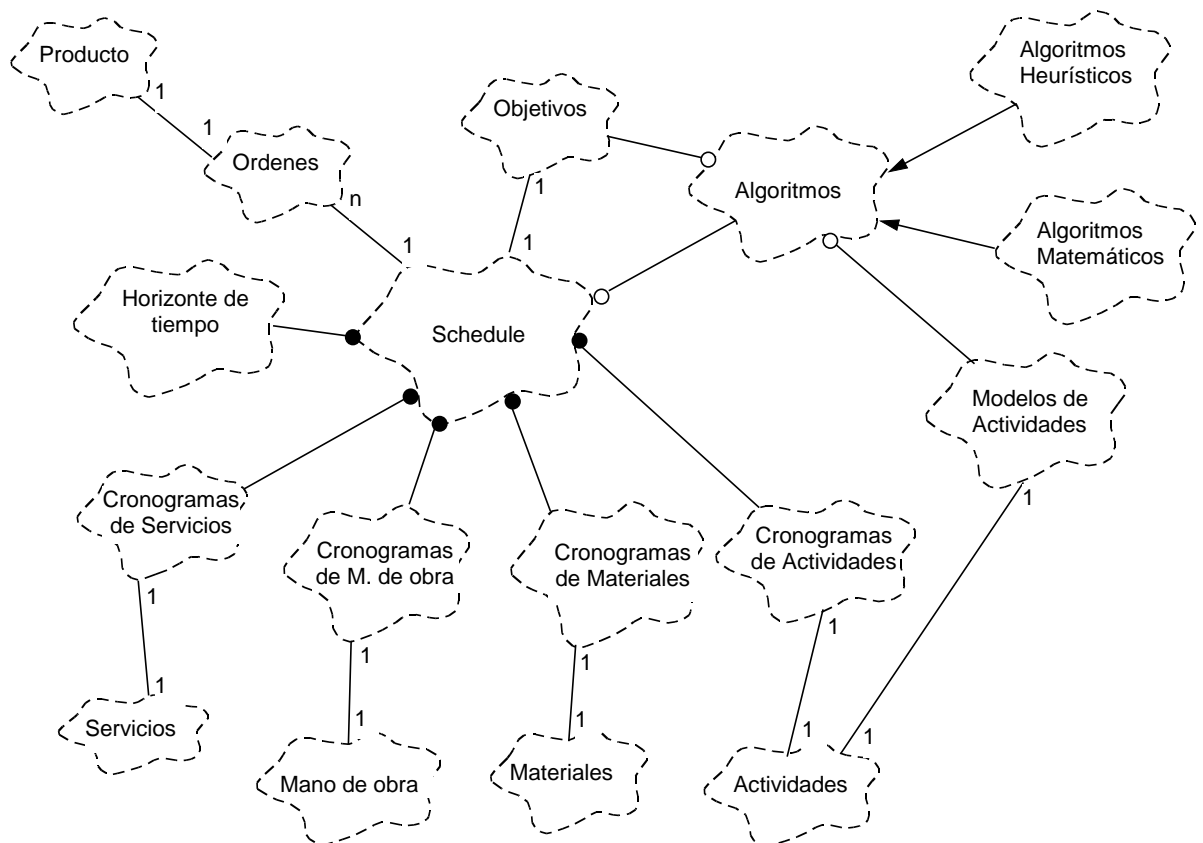


Diagrama de Clases 6

El problema de scheduling es un problema de optimización combinatoria, por esa razón, cuando el número de órdenes a someter a scheduling en una planta de producción por lotes es grande, se convierte en un problema de gran escala. En este caso los modelos matemáticos no son los más apropiados y se recurre usualmente a modelos heurísticos. Es decir, para soportar la tarea de scheduling el sistema debe incorporar algoritmos matemáticos y algoritmos heurísticos. Esto induce la necesidad de incluir dos nuevas clases/objetos: *Algoritmos Heurísticos* y *Algoritmos Matemáticos*. Ambas clases se vinculan con la clase *Algoritmos* a través de una relación de herencia. La clase *Algoritmos* en este caso contiene la referencia de todas las "instancias" de cada una de las subclases que la heredan. Esto se representa en el Diagrama de Clases 6.

En cuanto al comportamiento de las clases especificadas, para obtener flexibilidad en las definiciones, se establece que las clases *Actividades*, *Modelos de Actividades*, *Algoritmos*, *Objetivos*, *Ordenes*, *Unidades de Tiempo* y *Horizonte de Tiempo* deben definirse cada una en forma individual, y luego utilizarse para conformar un *Schedule* en particular. Esto plantea la situación simple donde cada clase tendrá sus actividades de *crear* una nueva instancia, *modificar* una instancia existente o *borrar* una instancia existente.

A diferencia de las anteriores las "instancias" de las clases *Cronogramas de Actividades*, *de Materiales*, *de Mano de Obra* y *de Servicio*, son creadas a partir de una "instancia" de la clase *Schedule*. Estas clases también tendrán sus operaciones básicas de crear, modificar y borrar, pero las mismas son activadas a partir de un mensaje enviado por la clase *Schedule*. Esto es así porque un *Cronogramas de...* no existe si no hay un *schedule* que le dé origen.

Hemos presentado así una comprensión básica del problema. El paso siguiente fue realizar un análisis detallado de la estructura de cada clase y de sus responsabilidades (métodos). Para esto hemos desarrollado el *Diccionario de Clases* para cada una de las clases. A modo de ejemplo se muestra el diccionario de la clase *Schedule*.

NOMBRE	<i>Schedule</i>
DESCRIPCIÓN	<i>Esta clase instancia los schedules de producción generados por el usuario. Entre ellos se encuentra el schedule actual (vigente).</i>
PROPIEDADES	<i>nombre</i> <i>identificación</i> <i>descripción</i> <i>lista de órdenes</i> <i>lista de materiales</i> <i>lista de servicios</i> <i>mano de obra</i> <i>lista de actividades</i> <i>algoritmo</i> <i>objetivo</i> <i>horizonte de tiempo</i>
RESPONSABILIDADES	<i>crearse</i> <i>destruirse</i> <i>modificarse</i> <i>mostrarse</i> <i>mostrar lista de órdenes</i> <i>mostrar lista de materiales</i> <i>mostrar lista de servicios</i> <i>mostrar lista de actividades</i> <i>ejecutar algoritmo de scheduling</i>

A partir de este análisis detallado se obtuvo información sobre las relaciones estructurales necesarias, (por ejemplo, que la clase *schedule* necesitará de una lista de los materiales requeridos, una lista de actividades asociadas, una lista de servicios y un horizonte de tiempo), como así también un detalle de los atributos y los métodos requeridos por cada clase.

BASE DE MODELOS

Los algoritmos de scheduling conforman esencialmente la base de modelos de decisión del subsistema de scheduling. La estructura del sistema es general y permite incorporar distintos algoritmos de scheduling. En esta primer etapa de desarrollo del prototipo hemos incorporado un algoritmo de tipo heurístico especialmente diseñado (Rossi y otros, 1997). El mismo consiste de un proceso iterativo que consta esencialmente de tres etapas:

- • *Secuenciamiento* del conjunto de órdenes a someter a scheduling.
- *Asignación* de cada orden a las actividades requeridas por el proceso de producción.
- *Evaluación* del cronograma resultante de asignar una secuencia.

Secuenciamiento: esta etapa establece un orden para asignar cada proceso de producción a las actividades requeridas por el mismo. La estrategia desarrollada realiza el secuenciamiento utilizando un *algoritmo genético* (Morton, 1994), que incluye el algoritmo GRASP para seleccionar la población inicial, y algoritmos alternativos (cruzamiento de pares adyacentes y PMX) para producir la evolución.

Asignación de una secuencia: utilizamos un mecanismo de asignación por proceso. La asignación por proceso puede variar según la política de inventario seleccionada: para el caso de admitirse inventario entre actividades, una actividad asociada a un proceso se asigna para ser iniciada apenas sea posible. Para realizar esta asignación hemos diseñado el algoritmo AFL (Asignación Forward por Lote) el cual realiza la asignación mediante una estrategia de avance. En el caso de no admitirse *inventarios intermedios*, la asignación es un poco más compleja. Para esto se utiliza un algoritmo de avance (admitiendo inventarios intermedios) para determinar la fecha de finalización. Luego partiendo de esta fecha se usa un algoritmo de retroceso (backward) sin admitir inventarios intermedios.

Evaluación: La evaluación consiste en determinar la performance (fitness) de una solución. El factor de performance de una solución queda determinado por el objetivo fijado por el decisor (Por ejemplo: minimizar Tiempo Total, minimizar Tardanza Total, minimizar Tardanza Media, minimizar Tardanza Máxima, minimizar Costo, etc.).

Si bien por el momento sólo hemos incluido un tipo de algoritmo de scheduling, el sistema ha sido estructurado de manera tal que nuevos algoritmos pueden ser incorporados a la base de modelos del SSS.

CONCLUSIONES

Como conclusión del presente trabajo debemos destacar: en primer lugar, conforme a lo propuesto por Sage 1991, que hemos adoptado una política de desarrollo evolutiva del SSD prototipo. Por esto, tanto el diseño de la estrategia de manejo de la información, como el análisis y diseño del subsistema fue avanzado hasta un nivel considerado satisfactorio y luego pasamos a la etapa de programación. Los resultados de esta primer versión académica del prototipo posiblemente sugieran cambios tanto en la estructura de la información como del diseño del sistema. En este trabajo hemos descripto la estrategia de soporte de decisión diseñada, los pasos seguidos y los criterios de análisis utilizados.

En segundo lugar, hemos centrado la descripción del subsistema exclusivamente en la información asociada al *schedule* y su generación. La información relativa a la capacidad de producción, la planificación de los materiales, la administración de las órdenes de producción, la estructura de los productos (lista de materiales y proceso de producción) y los requerimientos de recursos, necesaria para realizar el scheduling es responsabilidad de otros

subsistemas del SSD en desarrollo.

En tercer lugar, el algoritmo de scheduling incorporado a la base de modelos del sistema, permite alcanzar soluciones con buena performance en tiempos reducidos, además es muy "robusto" ya que no está asociado a modelos de procesos, a políticas de operación, ni a una función objetivo en particular.

Finalmente, la estructuración de la información del proceso a través de actividades ha permitido lograr una arquitectura muy flexible, ya que no está vinculada a un tipo de proceso, a un tipo de organización, ni a un tipo de producto en particular.

El prototipo se implementa utilizando el lenguaje de programación visual DELPHI, para lo cual debimos compatibilizar el diseño de objetos realizado siguiendo la metodología de G. Booch, y el lenguaje de programación. La adaptación más importante ha sido quizás la necesidad de almacenar objetos en tablas relacionales.

REFERENCIAS

- Adamopoulos G., N. Karacapilidis & S. Pantazopoulos, (1994), *Production Management in the Textil Industry Using the "YFADI" Decision Support System*, Comp. & Chem. Engng Vol **18**, pp s577-s583.
- Booch, G., (1993), *Object Oriented Analysis and Design*, J.Wiley & Sons.
- Crooks, C.A., K. Kuryian and Macchietto (1992), *Integration of discontinuous Plant Design, Automation and Operation Softwares Tools*, Computers & Chemical Engineering, 16S, pp S289-S296.
- Lorino, P. (1993), *El Control de Gestion Estratégico. La Gestión por Actividades*. Macombo, Barcelona.
- Morton, T.E. y D.W. Pentico, (1993), *Heuristic Scheduling Systems*, J. Wiley & Sons, New York
- Rico M., O. Yuschak, M.L. Taverna, J. C. Ramos, M.R. Galli and O. Chiotti, (Marzo 1997), *Decision Support Systems Generator for Industrial Companies*, Proceeding of 21th International Conference on Computers & Industrial Engineering. Puerto Rico.
- Rossi, P., (1997), M. Rico, M. R. Galli y O. Chiotti, *Scheduling de Actividades en un Proceso de Producción por Lotes*. 26 JAIIO.
- Sage, A.P., (1991), *Decision Support Systems Engineering*, John Wiley & Sons.